AMENDMENTS TO THE CLAIMS

LISTING OF CLAIMS

1. (currently amended) An updating system for transforming a first data image into a second data image, wherein said first image resides across k memory blocks of a block-structured non-volatile memory device contained in a client device, said updating system comprising:

a. An update generator that produces an update package resulting from a comparison between the first data image and the second data image whereby said comparison selects and encodes an instruction set comprising a plurality of SETBLOCK, COPY and ADD operations for each of the k memory blocks; and

b. An update decoder resident on the client device, whereby said update decoder interprets the instruction set of the update package and applies the update package to update the k memory blocks;

wherein the SETBLOCK operations identify operations applicable to specific memory blocks and facilitate memory block updating in a non-sequential manner.

2. (previously presented) The system of claim 1 further comprising a communications network and a host server that comprises the update generator, whereby the update package is delivered from the host server to the client device via the communications network.

3. (previously presented) The system of claim 1 wherein for each memory block X of k blocks an updated version of such Xth memory block is first constructed in a scratch memory, and then memory block X is reprogrammed with the contents of the scratch memory.

4. (previously presented) The system of claim 1 wherein said update package further includes a status array comprised of at least two switchable status identifiers associated with each memory block X of k to be updated as instructed by the instruction set contained in the update package.

5. (previously presented) The system of claim 3 wherein said k memory blocks are updated in a non-sequential order as specified by the SETBLOCK operations comprising the instruction set.

6. (previously presented) The system of claim 3 wherein the instruction set further comprises a plurality of COPYADD operations in lieu of at least a portion of the plurality of COPY operations.

7. (previously presented) The system of claim 3 wherein said update decoder maintains at least two copy-offset values comprised of a current offset value and a non-current offset value, and wherein the instruction set of said update package further comprises a plurality of SETCOPYOFFSET operations, which SETCOPYOFFSET instructions are instruct the update decoder to toggle the copy-offset value between the current value and the non-current value.

8. (previously presented) The system of claim 7 said update decoder further comprises a mode mechanism that switches the update decoder from using the copy-offset value to using a zero offset.

9. (currently amended) A method of updating to a second image a first image stored across k memory blocks of a non-volatile memory device contained in a client device, said updating method comprising:

a. Generating an update package by comparing the first image and the second image and using result of said comparison to encode an instruction set comprised of a plurality of SETBLOCK, COPY and ADD operations for each of the k memory blocks; and

b. Applying the instruction set by interpreting the instruction set to direct the updating of the memory blocks in an order specified by the SETBLOCK operations;

wherein the SETBLOCK operations enable memory block updating in a non-sequential manner.

10. (previously presented) The method of claim 9 wherein said applying step further comprises, for each memory block X of k blocks,

a. constructing an updated version of such Xth memory block in a scratch memory location accessible to the client device wherein said scratch memory location being at least as large as the largest of the k memory blocks, and

b. reprogramming Xth memory block with the contents of the scratch memory.

11. (previously presented) The method of claim 9 wherein the step of applying instruction set in an order specified by the SETBLOCK operation is a non-sequential order.

12. (previously presented) The method of claim 9 further comprising the steps of:

a. Constructing an updated version of each Xth memory block in a scratch memory location accessible to the client device wherein said scratch memory location is at least as large as the largest of the k memory blocks;

b. Reprogramming a temporary memory block in a non-volatile memory device with the contents of the scratch memory;

c. Switching a first switchable status identifier when step b is completed with respect to each memory block;

d. Reprogramming the Xth memory block with the contents of the temporary memory block; and

e. Switching a second switchable status identifier when step d is completed with respect to each memory block.

13. (previously presented) The method of claim 12 further comprising the steps of:

Checking the second status identifier for each Xth block;

Then for the first block encountered for which such second status identifier is still set, checking the first status identifier with respect to such; and

Proceeding to update such block commencing from step (d) above if said first status identifier is clear, or proceeding to update the subsequent block X+1 commencing at step (a) above.

14. (previously presented) The method of claim 9 further comprising the steps of:

a. Maintaining a first copy-offset value and a second copy-offset value,

b. Selection Setting a copy-offset value by a plurality of COPYOFFSET values in the client.

15. (currently amended) A system for reliably updating on a client device a first image stored across a plurality of memory blocks of a non-volatile memory device to create a second image, said system comprising:

a. an update package including an instruction set, which instruction set comprises a plurality of ADD and COPY operations associated with each of the plurality of memory blocks to be updated;

b. a status array comprised of a least two switchable status identifiers associated with each of the plurality of memory blocks, wherein one X block of k blocks to be updated as instructed by the instruction set is contained in the update package; and

c. an update decoder resident on said client device that interprets the update package and applies the instruction set to update the plurality of blocks on a block-by-block basis, and which update decoder accesses and manipulates the status identifiers when applying said instruction set;

wherein said at least two switchable status identifiers are configured to facilitate restarting update processing if said update processing is interrupted.

16. (currently amended) The system of claim 15 further comprising a status array of at least first and second switchable status identifiers associated with each Xth memory block of k to be updated, whereby the ~~second~~ first switchable status identifier is switched from a first state to a second state when the contents of the scratch memory are stored in a temporary memory block in a non-volatile memory device prior to said contents being reprogrammed into the Xth memory block, and whereby the ~~first~~ second switchable status identifier is switched from a first state to a second state when the updated code is reprogrammed into the Xth memory block from said temporary memory device rather than with the contents of the scratch memory [status bit].

17. (previously presented) The system of claim 15 wherein said status array is comprised in the update package.

18. (previously presented) The system of claim 17 wherein the instruction set further comprises a plurality of SETBLOCK operations, at least one each associated with each of the plurality of memory blocks to be updated.